What is claimed is:

1. A server for providing data to clients, the server comprising:

an OSI layer 4 dispatcher having a queue for storing connection requests received from clients; and

at least one back-end server;

wherein the dispatcher stores in the queue one or more of the connection requests received from clients when the back-end server is unavailable to process said one or more connection requests;

wherein the dispatcher retrieves said one or more connection requests from the queue for forwarding to the back-end server when the back-end server becomes available to process said one or more connection requests; and

wherein the dispatcher determines whether the back-end server is available to process said one or more connection requests by comparing a number of connections concurrently supported by the back-end server to a maximum number of concurrent connections that the back-end server is permitted to support, the maximum number being less than a maximum number of connections which the back-end server is capable of supporting concurrently.

2. The server of claim 1 wherein the dispatcher is configured to monitor a performance of the back-end server, to define the maximum number of concurrent connections that the back-end server is permitted to support, and to dynamically adjust the maximum number in response to the monitored performance.

3. The server of claim 1 wherein the server is a cluster-based server comprising a plurality of back-end servers, wherein the dispatcher is configured to store in the queue said one or more connection requests when none of the back-end servers is available to process said one or

more connection requests, and wherein the dispatcher is further configured to retrieve said one or more connection requests from the queue for forwarding to one of the back-end servers when said one of the back-end servers becomes available to process said one or more connection requests.

4. The server of claim 1 wherein the server is a Web server.

5. The server of claim 1 wherein the dispatcher and the back-end server are embodied in COTS hardware.

6. The server of claim 1 wherein the dispatcher comprises a first computer device, wherein the back-end server comprises a second computer device, and wherein the first and second computer devices are configured to communicate with one another over a computer network.

7. A method for controlled server loading, the method comprising:

receiving a plurality of connection requests from clients;

establishing, in response to some of the connection requests, a number of concurrent connections between a server and clients; and

storing at least one of the connection requests until one of the established connections is terminated.

8. The method of claim 7 wherein the number of concurrent connections established between the server and clients is less than a maximum number of connections which the server is capable of supporting concurrently.

9. The method of claim 7 further comprising retrieving the stored connection request after at least one of the established connections is terminated, and establishing a connection between the server and a client associated with the retrieved connection request.

10. The method of claim 7 wherein the storing includes storing a plurality of the connection requests, the method further comprising retrieving one of the stored connection requests and establishing a new connection between the server and a client associated with the retrieved one of the connection requests each time one of the established connections is terminated.

11. The method of claim 10 wherein the retrieving includes retrieving the stored connection requests on a FIFO basis.

12. The method of claim 7 wherein the connection requests are TCP requests.

13. The method of claim 7 wherein at least the receiving and the storing are performed by a single computer device having at least one processor.

14. The method of claim 13 wherein the single computer device comprises the server.

15. A computer-readable medium having computer-executable instructions for performing the method of claim 7.

16. A method for controlled server loading, the method comprising:

defining a maximum number of concurrent connections that a server is permitted to support;

monitoring the server's performance; and

dynamically adjusting the maximum number in response to the monitoring to thereby adjust the server's performance.

17. The method of claim 16 wherein the monitoring includes monitoring the server's performance in terms of a performance metric selected from the group consisting of average response time, maximum response time, and server packet throughput.

18. The method of claim 16 further comprising receiving a plurality of connection requests from clients, establishing in response to some of the connection requests the maximum number of concurrent connections with the server, and storing at least one of the connection requests until one of the established connections is terminated.

19. The method of claim 18 wherein the dynamically adjusting includes dynamically adjusting the maximum number as a function of the number of connection requests that are concurrently stored.

20. The method of claim 19 wherein the dynamically adjusting includes increasing the maximum number when the number of concurrently stored connection requests is greater than a predefined number.

21. The method of claim 19 wherein the dynamically adjusting includes decreasing the maximum number when the number of concurrently stored connection requests is less than a predefined number.

22. A method for controlled loading of a cluster-based server, the cluster-based server including a dispatcher and a plurality of back-end servers, the method comprising:

receiving at the dispatcher a plurality of connection requests from clients;

forwarding a plurality of the connection requests to each of the back-end servers, each back-end server establishing a number of concurrent connections with clients in response to the connection requests forwarded thereto; and

storing at the dispatcher at least one of the connection requests until one of the concurrent connections is terminated.

23. The method of claim 22 wherein the storing includes storing a plurality of the connection requests, and wherein the forwarding includes forwarding one of the stored connection requests to one of the back-end servers each time one of the concurrent connections is terminated.

24. The method of claim 22 wherein the cluster-based server is an L4/3 server.

25. A method for controlled loading of a cluster-based server, the cluster-based server including a dispatcher and a plurality of back-end servers, the method comprising:

defining, for each back-end server, a maximum number of concurrent connections that can be supported;

monitoring the performance of each back-end server; and

dynamically adjusting the maximum number for at least one of the back-end servers in response to the monitoring to thereby adjust the performance of the cluster-based server.

26. The method of claim 25 wherein the dynamically adjusting includes dynamically adjusting the maximum number for each back-end server.

27. The method of claim 25 further comprising receiving a plurality of connection requests from clients, forwarding some of the connection requests to the back-end servers, each back-end server establishing a number of concurrent connections with clients in response to the connection requests forwarded thereto, and storing at least one of the connection requests until one of the concurrent connections is terminated.

28. The method of claim 27 wherein the dynamically adjusting includes dynamically adjusting the maximum number

for said one of the back-end servers as a function of the number of connection requests that are concurrently stored.